National College of Ireland

Assessment of Alzheimer image detection on CNN ensemble model fine-tuned with genetic algorithm

MSc Research Project
MSc in Artificial Intelligence

Carlos Angel Herrera Padilla
Student ID: x21148414

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Carlos Angel Herrera Padilla
Student ID:	x21148414
Programme:	MSc in Artificial Intelligence
Year:	2023
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	31/01/2024
Project Title:	Assessment of Alzheimer image detection on CNN ensemble model fine-tuned with genetic algorithm
Word Count:	6809
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	30th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Assessment of Alzheimer image detection on CNN ensemble model fine-tuned with genetic algorithm

Carlos Angel Herrera Padilla
x21148414

Abstract

Alzheimer's is a brain disease that affects millions of people around the world, by 2060 it is estimated that more than 14 million people in the United States will be diagnosed with this disease. The importance of detecting Alzheimer's in an individual could change their life, even though Alzheimer's is a disease with no cure, the detection of Alzheimer's could delay or slow the symptoms, one of the many tasks that machine learning is exceeding is in classification problems. Convolutional Neural Network (CNN) models are commonly used to classify images in the health industry. For this reason, a simple custom CNN model is developed so that it goes through fine-tuning with a genetic algorithm, making an ensemble model of the three best models of the genetic algorithm. This project assesses the performance of the proposed model against state-of-the-art pre-trained ensemble models. The proposed model had an Area Under the Receiver Operating Characteristic Curve (ROC AUC) score of 0.773 and has the third-best performance against ensemble models of pre-trained models like the VGG16, Inceptionv3, and the ResNet50.

1 Introduction

1.0.1 Background and Motivation

Alzheimer's is a brain disorder that affects the memory and thinking of the individuals who have it. It is estimated that more than 14 million people will be diagnosed with Alzheimer's in the United States by 2060[1]. Currently, Alzheimer's disease has no cure, and the early detection of this disease could be very important to delay and slow the symptoms. Alzheimer's disease is on the top ten leading causes of death in the United States[2], so it is important to use every tool in our disposition to try to detect this disease as early as possible and one of these tools is machine learning. Machine learning algorithms like deep learning could help to detect Alzheimer's in a very early stage and help to slow and delay the disease. CNNs are commonly used to help with classification problems but most of the current CNN models are pre-trained models that use the transfer learning method it is an excellent way to solve these classification problems, but they require a high computational power to use them, and retrain them to solve specific problems. This is why this project would try to develop and implement a simple CNN model that will be passed to a genetic algorithm to find the best parameters for the model, creating an

[1] `https://www.cdc.gov/aging/aginginfo/alzheimers.htm`
[2] `https://www.cdc.gov/aging/aginginfo/alzheimers.htm`

ensemble model of the best models that the genetic algorithm can find, and we are going to evaluate the model against other ensemble models made of pre-trained state-of-the-art models. The goal of this project is to develop a simple ensemble model that can classify as well as state-of-the-art pre-trained models but without the need to require high computational power like the pre-trained models have, so that the proposed model can be implemented anywhere without restrictions and can help in classification problems just like detecting Alzheimer's in an individual.

1.0.2 Research Question

This project pursues an answer to the following research question: How well can a custom CNN ensemble model fine-tuned by a genetic algorithm perform against state-of-the-art pre-trained ensemble models on image classification for Alzheimer's?

2 Related Work

2.1 CNN Approaches on Alzheimer's Classification

Khagi et al. (2019) proposed a CNN architecture of 22 layers and compared the proposed model to three pre-trained models that were fine-tuned to the Open Access Series of Imaging Studies (OASIS) dataset. The proposed architecture of the authors consisted of four convolutional layers, two convolutional layers with 64 filters, and two convolutional layers with 32 filters. After each convolutional layer the authors put a batch normalization layer followed by an activation function layer, the authors proposed a Rectifier Linear Function (ReLu) followed by a pooling layer using a max pooling function, connected to a 512 fully connected layer with another activation function of ReLu to connected to another fully connected layer of 2 neurons with a SoftMax function. The authors compared the proposed model against the AlexNet, GoogleNet, and ResNet50 pre-trained models, showing a very good training accuracy of 98% against 94% of the ResNet50 model, 89% of the GoogleNet model, and 94% AlexNet model. It is important to note that even though the proposed model had a very good performance the authors only selected a dataset with 1,680 images, 840 images for patients with Alzheimer's, and 840 images for patients without Alzheimer's. The selection of a dataset with a distribution of 50% of the images for each class does not represent real-world data and the proposed model of the authors could potentially have a bad performance with real-world data.

2.2 Transfer Learning Approaches on Alzheimer's Classification

Ebrahim et al. (2020) explored how the division of the images on the dataset for training and testing can affect a model. The authors have used the VGG16 pre-trained model to make use of the transfer learning technique and take advantage of the feature selection that the VGG16 already has, and they retrained for detecting Alzheimer's. The dataset that the authors have used contains 2560 images for persons with no Alzheimer's and 2519 images for persons with Alzheimer's. The authors created three splits of training and testing. The first split was 20% testing and 80% training having an accuracy of 95.3%, the second split was 30% testing and 70% training having an accuracy of 95.7%, and the third and last split was 40% testing and 60% training having an accuracy of 93.6%. From the results, we could appreciate how splitting the data into training and

testing can affect the overall accuracy of the model. One thing to take into account is that the total amount of images is not enough to have a robust model and as shown in Khagi et al. (2019) having almost the same number of images for both cases does not represent real-world data and the proposed model could not be well suited for real-world data. Another thing that the paper is missing is the comparison with other models, comparing the model that the authors have proposed with different models can give us a look to see if the proposed model of the authors is good, and having more models should be helpful to see if the division of different training and testing split affects the other models the same way it affected the authors model.

Oktavian et al. (2022) explored another transfer learning method that uses the ResNet-18 and takes advantage of the pre-training that the model has with the ImageNet dataset. The authors proposed a model with weighting the loss function. The dataset the authors have used is the Alzheimer's Disease Neuroimaging Initiative (ADNI) which contains 133 images of patients with mild cognitive impairment, 58 with Alzheimer's disease, and 115 with normal controls. The original dataset contained the images in a 3-dimensional NII format, so the authors sliced every image into 256 slices, resulting in 10,794 total 2-dimensional images. The proposed model of the authors achieved an accuracy of 88.30% in multi-class classification, achieving higher than other models like the VGG-16 with 85% accuracy and the ResNet-50 with 75% accuracy, but it did not improve the accuracy of the AlexNet model with a 96% accuracy. Even though the author's model was not the best overall, they demonstrated a good model with a good approach by weighting the loss function.

Abed et al. (2020) explored how good pre-trained models can be to classify if a patient has Alzheimer's or not. The authors proposed using pre-trained models for this task, and they selected three pre-trained models: VGG19, ResNet50, and InceptionV3. The authors used the ADNI dataset as well as it happened in Oktavian et al. (2022), the authors had to process 3-dimensional images into 2-dimensional slices and distributed the resulting 50,000 images into a train-test split of 80% of the images for training and 20% of the images for testing. The authors trained again with the extracted dataset of the three pre-trained models, they made use of the weights the pre-trained models had with the ImageNet dataset, and it showed very good results with the VGG-19 achieving the best results with an accuracy of 93%, the Inceptionv3 model achieved an accuracy of 89% and lastly the ResNet50 with an accuracy of 85%.

Jabason et al. (2019b) proposed a method to train an ensemble model based on an entropy approach. Normally the training methods technique most of the CNN models use is selecting the images for training at random but the authors explored selecting the training images by the entropy value of the image, so the training image set had the images that have the most information. The authors have used the OASIS dataset. This dataset is a 3-dimensional dataset, so the authors had to process the images to have 2-dimensional slices. The CNN model that the authors proposed is using a pre-trained DenseNet model to extract the features and then pass it to a Long Short-Term Memory (LSTM) so then it can classify the images. The authors have used three different DenseNet models: the DenseNet121, the DenseNet169, and the DenseNet201. Of the three DenseNet models, the best accuracy is the DenseNet201 with 98.77%, followed by the DenseNet169 with an accuracy of 97.52%, and finally the DenseNet121 with an accuracy of 95.56%. As we can observe from the author's results the deeper the architectures of the pre-trained models the better the results are while classifying.

2.3 Ensemble Learning Approaches on Alzheimer's Classification

Another method to use pre-trained models is using the ensemble method, this method combines different models to give a classification. Francis and Pandian (2021) proposed an ensemble model of two pre-trained models: the Xception model and the MobileNet model. The authors used more than 100,000 features of the Xception model and more than 50,000 features of the MobileNet model extracted from previous training of the models, to combine the two models and predict. As well as the papers presented above, the authors have used the ADNI dataset, and they transformed the 3-dimensional images into 2-dimensional images getting as a result 1,050 images in total. The authors used 900 images for training and 150 images for testing. The results of the ensemble model of the authors proved to achieve better performance than the individual pre-trained models, the ensemble model proposed achieved 91.3% accuracy while the Xception model achieved 89.23% accuracy and the MobileNet had 89.89% accuracy. Overall, the authors demonstrated the benefits of using ensemble models to achieve higher performance in classification tasks.

Jabason et al. (2019a) proposed an ensemble method of a combination of three ensemble models. The authors used the OASIS dataset, transforming the 3-dimensional images into 2-dimensional representation slices. The innovation in the ensemble model that the authors proposed is in the classification of the training images, the authors separated the 2-dimension slices into three categories depending on the spatial information of the 3-dimensional representation of the images: axial, coronal, and sagittal. After the authors divided the images into these three planes they passed the images of each plane to an ensemble model, made of the DenseNet model and the ResNet model. The ensemble model for the three spatial planes is different, the authors then passed the results of each ensemble model onto the different planes and combined the results of the three models to create another ensemble and classify the images. The proposed model of the authors achieved 95.23% accuracy, getting the worst results that the same authors developed in the paper mentioned in the last section where they used a DenseNet model to extract the features of the same OASIS images and then passed it through an LSTM algorithm getting a result of 98.77% accuracy.

Sadat et al. (2021) have proposed an ensemble method of 6 different CNN models, the authors use the VGG-19, the Inception-ResNetv2, the ResNet152v2, the EfficientNetB5, the EfficientNetB6, and a custom model created by the authors. The dataset used for this paper is the OASIS-I dataset and one thing that the authors have done differently from other ensemble models is that they did a weighted average ensemble method where they gave higher weights to the EfficientNetB5 model, and the custom CNN model the authors proposed. The accuracy achieved for the ensemble model was 96%, getting the same results as state-of-the-art models like the InceptionV4 that achieved the same accuracy of 96%. One thing that the authors have done differently from other papers is that they present the F1-Score of the proposed model. This is a very good metric to evaluate a model because it combines the precision and recall metrics into one metric. The F1-Score is a good metric to show how well the model is classifying and it is especially good where the utilized dataset is imbalanced. The F1-Score of the proposed model is 0.95, this tells us that the model the authors proposed is very good and it can differentiate very well between a person with Alzheimer's and a person without Alzheimer's.

Loddo et al. (2022) explored the use of pre-trained models and ensemble techniques

to detect Alzheimer's from MRI images. The authors used three different datasets containing different stages of dementia, they used images for individuals with very mild dementia, moderate dementia, mild dementia, and no dementia at all. The authors proposed using an ensemble mode of three pre-trained models, the ensemble models of the authors consisted of an InceptionResNetV2, a ResNet-101 model, and an AlexNet model. The proposed ensemble model was compared to how well it performed against the previously mentioned models. The proposed ensemble model of the authors demonstrated an accuracy of 96.02% and it demonstrated to be the best performer model against the standalone pre-trained models.

Kang et al. (2021) proposed an ensemble method with slice selection to 3D volumes to detect Alzheimer's from images. The authors passed the sliced images to two pre-trained models to train them with the selected images. The models trained were the VGG16, the ResNet50, and the discriminator of an adversarial network. After the models were trained with the sliced images the authors created an ensemble model with the three models mentioned above and the authors used the ensemble model to see how well the proposed ensemble model could differentiate between normal individuals against individuals with Alzheimer's, normal individuals against people with mild cognitive impairment, and normal individuals against people with mild cognitive impairment. The proposed ensemble model demonstrated to have an accuracy of 90.4% against 90.36% for the discriminator adversarial algorithm, 87.95% for the VGG16, and 83.13% for the ResNet50 in differentiating between individuals with Alzheimer's and normal individuals. The accuracy of the proposed ensemble model for differentiating between individuals with Alzheimer's and individuals with mild cognitive impairment was 77.2% against 74.56% for the adversarial algorithm, 77.19% for the VGG16 and 74.56% for the ResNet50 model. The accuracy for differentiating between individuals with mild cognitive impairment against normal individuals was 72.4% for the proposed model against 69.11% for the discriminative adversarial algorithm, 69.92% for the VGG16 model, and 65.85% for the ResNet50 model. The proposed ensemble model of the authors achieved better results in all categories than all the individual models in the proposed ensemble model.

Islam and Zhang (2018) explored the use of ensemble models for detecting several stages of dementia. The authors proposed an ensemble model of three variants of convolutional neural networks, each of those convolutional neural networks will be slightly different but they all have the same architecture. The neural networks were composed of a convolutional layer, a batch normalization layer, a rectified linear unit layer, and a pooling layer. The authors then trained the models and performed an ensemble model with them, they used several versions, and, in the end, they ended up with three different versions for the ensemble models. The proposed ensemble model of the authors achieved 93% accuracy against 77% and 78% accuracy of the two other variants of the ensemble model. The proposed model demonstrated superior performance against pre-trained models like the Inception-v4 with 75% accuracy and the ResNet model with 82% accuracy.

Khoei et al. (2021) proposed a similar ensemble method that is proposed in this project. The authors have proposed an ensemble method of four different classifiers: the Random Forest, the Naïve Bayes, the Support Vector Machines (SVM), and K-Nearest Neighbors (KNN). This ensemble model will be fine-tuned with a genetic algorithm and then the authors will compare the proposed model with the performance of the individual models. The dataset used for this paper was the ADNI dataset, this dataset contains 8211 image samples. The proposed ensemble model achieved the best results with an

accuracy of 96.7% against the 96.3% accuracy of the Random Forest, the 82.6% accuracy of the Naïve Bayes, the 89.9% accuracy of the SVM, and the 91.9% accuracy of the KNN. In terms of the F1-Score the proposed model also performed better than the individual models achieving an F1-Score of 0.971 against the 0.97 of the Random Forest, the 0.848 of the Naïve Bayes, the 0.921 of the SVM, and the 0.94 of the KNN. This paper uses the genetic algorithm to fine-tune the parameters of the ensemble model, this technique is similar to the technique that this project tries to implement but the difference resides in that this project will use the genetic algorithm to fine-tune the parameters of a custom CNN model to find the best three CNN models with the best parameters to create an ensemble model of those three models and compared the result with state-of-the-art ensemble pre-trained models, not just comparing them with the individual models like this paper did.

3 Methodology

The methodology for this project consists of seven stages: data gathering, data pre-processing, data augmentation, CNN architecture, fine-tuning of the CNN with a genetic algorithm, training and testing, and creation and assessment of an ensemble model. In Figure 1 we can see the proposed methodology.

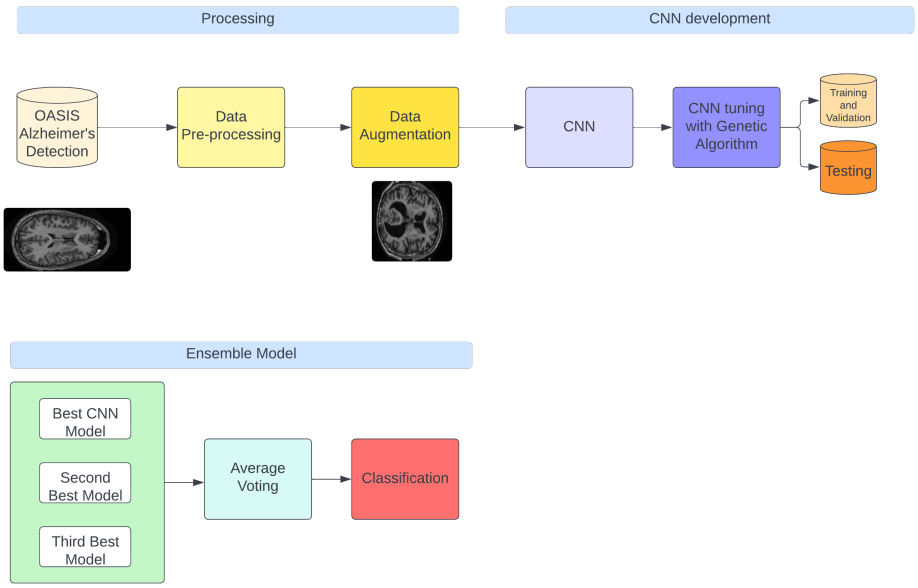

Figure 1: Project Methodology

3.1 Data Gathering

The dataset used for this project is the OASIS Alzheimer's Detection[3] dataset. This dataset contains 86,437 images of 496x248 pixels from 461 patients containing four classes: non-demented, mild dementia, moderate dementia, and very mild dementia. This dataset

[3] https://www.kaggle.com/datasets/ninadaithal/imagesoasis

was extracted originally from the OASIS MRI[4] dataset but the images are processed in the JPG format instead of the original NII format that is commonly used for storing MRI data and these types of files are more difficult to manipulate in machine learning applications and that is why the OASIS Alzheimer's Detection dataset was used instead of the original OASIS MRI dataset. In Table 1 we can see the OASIS Alzheimer's Detection dataset.

Table 1: OASIS Alzheimer's Detection dataset.

Class	**Subclass**	**Images**
Positive	Mild Dementia	5,002
	Moderate Dementia	488
	Very Mild Dementia	13,725
Negative	Non Demented	67,222

3.2 Data Pre-processing

The scope of the project is to classify correctly if an individual has Alzheimer's regardless of how advanced the dementia stage is, therefore, the Oasis Alzheimer's Detection dataset was preprocessed to gather all the types of dementia together in one single class, the positive class. In Table 2 we can see the final dataset used for the project.

Table 2: Final Dataset.

Class	**Total Images**
Positive	19,215
Negative	67,222

After merging all the instances of dementia to have only the Positive and Negative classes, the dataset was then split into training, validation, and testing sets. The images were divided into 75% for training, 20% for testing, and 5% for validation.

From Table 2 we can observe that the dataset is completely unbalanced with the Negative class having almost 4.6 times more images than the Positive class, for this reason, a data augmentation needs to be done so we can balance the dataset and the models can predict more accurately.

3.3 Data Augmentation

Data augmentation is a technique that helps to increment the training data by creating new images with the existing data[5]. This technique is important because the training data becomes more balanced, so the model can classify better, and it will also help to prevent over-fitting when training. Image augmentation can be done in several forms, by randomly flipping the image, rotating the image, stretching the image, zooming the image, or cropping the image.

[4] `https://www.oasis-brains.org/`

[5] `https://www.datacamp.com/tutorial/complete-guide-data-augmentation`

4 Design Specification

In this section, we explain the concepts and techniques that were used for the development of the models used in this project.

4.1 Convolutional Neural Networks

Deep Neural Networks (DNN) are widely used to solve problems in computer vision, in particular, CNNs. CNN algorithms are specifically designed to work with images and using this type of algorithm for medical purposes has demonstrated great results. The CNN architecture can be as simple or as complicated as we want, the most basic CNN architecture consists of three layers: the convolutional layer, the pooling layer, and the fully connected layer.

4.1.1 Convolutional Layer

The mathematics technique of convolution takes place in this layer, this technique is just a dot product between two matrices. The first matrix consists of a kernel that will be moving around an image, and it will produce the dot product every time it moves through the image, making the activation map, which is just a representation of the original image. The convolution technique introduces linearity to the CNN model, so sometimes it is useful to put a non-linearity function after this layer. In this layer, almost all the computational power required in a CNN model takes place.

4.1.2 Pooling Layer

This layer reduces the spatial dimensions of the activation map created in the convolutional layer. The pooling operation derives a statistical summary of the outputs in the activation map.

4.1.3 Fully Connected Layer

This is the final layer of the CNN model, in this layer, all the neurons created before are connected with all the layers, and each one of the connections of neurons has an assigned weight.

4.2 Genetic Algorithm

The genetic algorithm is an algorithm inspired by the natural evolution theory of Charles Darwin. This algorithm imitates the natural selection process, this kind of algorithm is commonly used for problems in optimization and searching. This type of algorithm consists of five stages: initial population, fitness function, selection, crossover, and mutation.

4.2.1 Initial Population

In this stage, all the initial parameters must be initialized. Different types of parameters like the number of generations, population, and mutation rate can be initialized, as well as the parameters that we want to change in the CNN model, like the activation functions, the optimizer functions, or the size of the hidden layers.

4.2.2 Fitness Function

The fitness function is the metric in which the genetic algorithm will evaluate every individual of each generation. In the case of this project, the fitness function will be the F1-Score. The F1-Score was selected because we have an imbalanced dataset and even though we did data augmentation to correct this issue, it is better to use the F1-Score to see how well the model can classify between the positive and negative classes.

4.2.3 Selection

After the genetic algorithm evaluates every individual in the current generation with the fitness function, the two top performers will be chosen, and they will act as parents for the next generation.

4.2.4 Crossover

When the genetic algorithm finishes choosing the best two performing individuals in the generation, a crossover will occur. The best attributes of the two individuals will be passed on to the next generation.

4.2.5 Mutation

To ensure that the next generation still maintains diversity, a random mutation can occur in the parameters of the individuals in the generation. These mutations are small but, they are enough to ensure that the generation still has the diversity of individuals to find the best solution.

We are going to use this type of algorithm to optimize the CNN model and extract the three best individual models of the last generation of the genetic algorithm to create the ensemble model.

4.3 Transfer Learning

Transfer learning is a machine learning method that utilizes pre-trained models to be reused in a task that is different from what it was originally trained. This method is good for problems in computer vision because you can take the knowledge in a model and apply it to a new problem without the need to train the model from scratch, saving a lot of time and computational power. The advantage of this method is that the models are generally trained with huge datasets like the ImageNet dataset which contains more than 14 million images[6], and it means that the models trained with this dataset have a huge advantage over individual models because all the data and features than the model extracts on that kind of datasets.

4.4 Ensemble Models

The creation of ensemble models is a machine learning technique that combines multiple individual models to help in the classification process and it increases the performance of a task. This technique works well because it combines the strengths of each of the models that are combined and the result that it produces is a model that is more robust and

[6] `https://paperswithcode.com/dataset/imagenet`

accurate than the individual models. There are different ensemble techniques but the one that is utilized in this project is the average technique. As its name says, this technique is implemented by aggregating the output of each one of the models and getting the average. This technique works well in this approach because it can be used with different kinds of models which helps with the diversity of the final ensemble model, it also tends to produce more reliable and stable classifications while also reducing the computational power needed. As we will see below, several ensemble versions of pre-trained models were created to assess how well the proposed ensemble model can perform against state-of-the-art pre-trained ensemble models.

5 Implementation

In this section, all the models used in this project were implemented in a Python environment with different libraries. The most important libraries for this project are the Tensorflow library and the Keras library, with these two libraries, the loading and processing of the data were done as well as the creation, training, and testing of the proposed model along with the pre-trained models chosen to assess the performance of the proposed model. Before coming to the final proposed model, the implementation of the CNN model passed through several versions, every version of the CNN model several pre-trained models were implemented with the same conditions regarding the data handling the CNN model has, so we can compare how the proposed model is performing and how it affects on the model how the data is managed.

5.0.1 CNN Architecture

The initial proposed CNN Architecture consists of 11 layers: three convolutional layers, three activation layers, three max-pooling layers, one flattened layer, and finally one dense layer. From this architecture, the model is the base that every single one of the versions considers and then changes depending on the CNN model version.

5.0.2 CNN Version 1

The first version of the CNN model consists of the 11 layers mentioned above. The model has a kernel size of 3x3, and the first convolutional layer has 32 filters, the second convolutional has 64 filters and the third convolutional layer has 128 filters. After every one of the convolutional layers, an activation function is inserted, this is because the convolutional layer is a linear operation and our classification problem is not linear, so we introduce a non-linear function and the chosen function is the ReLu, this function was used because it is a function that is efficient to compute, and it introduces sparsity into the model. After the activation functions a max pooling function with size 2x2 is set, this will help in extracting the features of the images and it will reduce the computational complexity. The last two layers of this version are the flatten layer and the dense layer. The flatten layer, as its name says, flattens the output of all the previous layers and prepares it for the dense layer. Finally, the dense layer is the fully connected layer that we talked about in section 4.1.3, this layer has two units which are going to be the Positive and Negative classes to classify if an individual has Alzheimer's or not, and it also has an activation function. The activation function that is chosen for the fully connected layer

is the sigmoid function. This function was used because it is a very good function for binary classification problems, which is the case in this project.

It is also good to note that the dataset used to train this CNN version and pre-trained models is the normal dataset without any data augmentation. The pre-trained models chosen to implement are the VGG16 model, the ResNet50 model, and the InceptionV3 model.

5.0.3 CNN Version 2

In this version, the same CNN model of version one is used but now we generate some data augmentation. The data augmentation utilized in this version contains a combination of deforming the image so the model can observe features from diverse angles, zooming the images at random so that the model can observe features at diverse scales, and flipping the images at random in a horizontal way so that the model can distinguish the features regarding the orientation of the images. This data augmentation is done because, as we saw in Figure 2, the dataset is imbalanced because the negative class has almost 3.5 times more data than the positive class. Data augmentation will help to introduce more images of the positive class.

Just as happened with version 1, all the pre-trained models were trained again but this time the dataset has been augmented.

5.0.4 CNN Version 3

In this version of the CNN model the core architecture of Versions 1 and 2 remains the same but we add some new layers. After each activation function, we introduce a batch normalization layer to help improve the model stability and generalization. This happens because batch normalization helps to reduce the noise that the model could learn in the training data. Another layer that was added in this version is the dropout. A dropout layer was added because it helps the model to prevent overfitting the dropout layer is dropping some neurons of the model at random so that the model can learn a more generalized representation of the data, this dropout layer is added after each max pooling layer. In Figure 2 we can see the final CNN architecture of the model.

In Version 2 we added some data augmentation but, in this version, we added one more thing to combat the imbalanced dataset, besides the data augmentation we introduce class weights[7]. We are going to assign weight to the positive and negative classes to give the positive class more importance during training, so it balances out the majority class which is the negative class. The weights for each class are calculated by the *compute_class_weight* function of the scikit-learn library.

Once again, all the pre-trained models were trained again but this time adding the class weights.

5.0.5 Fine-Tuning with Genetic Algorithm

The introduction of the fine-tuning of the CNN model will take place after observing which of the three CNN versions is performing the best. After the selection of the best CNN version, we introduce the genetic algorithm so it can give us the best parameters of the CNN model. The initial parameters of the genetic algorithm will have 5 generations with a

[7] `https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/#h-what-are-class-weights`

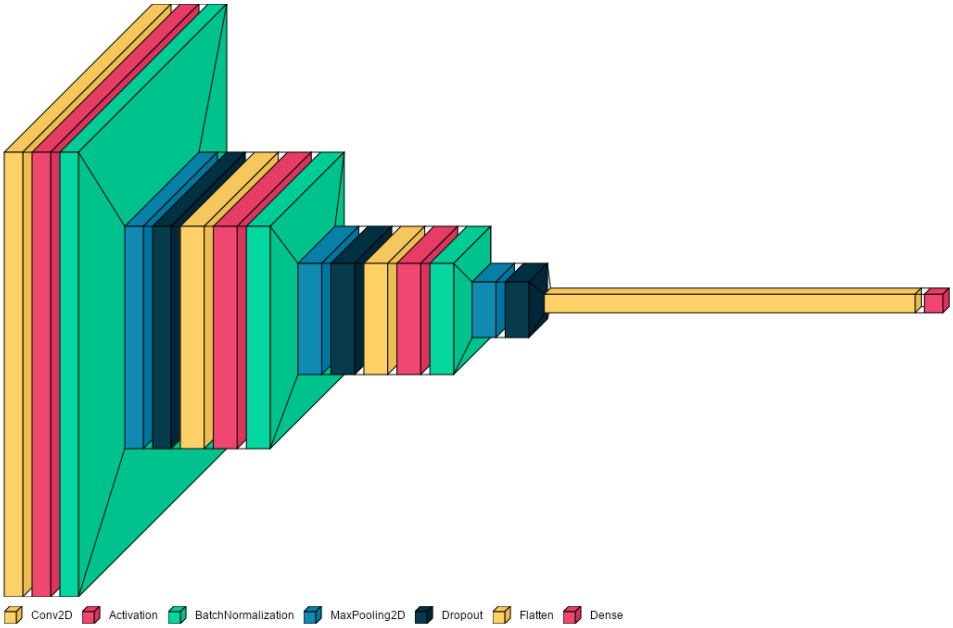

Figure 2: Final Architecture of the proposed CNN model

population of 10, and a mutation rate of 0.1. The parameters that the genetic algorithm will look to change in the CNN model are the size of the filter in each convolutional layer, the optimizer function, and the activation function. The possible sizes of the filter for the convolutional layers are 32, 64, 128, and 256. The possible optimizer functions are the Adam function, the Root Mean Squared Propagation (RMSprop) function, and the Stochastic Gradient Descent (SGD) function. The possible activation functions are the sigmoid function, the softmax function, and the hyperbolic tangent (tanh). Due to computational restraints, more parameters could be modified by the genetic algorithm.

5.0.6 CNN Ensemble Model

For the CNN ensemble model, we are selecting the three best models of the last generation of the genetic algorithm. We are selecting three models due to the fact that the computational power required is very high. After implementing the custom CNN ensemble model, we are implementing several ensemble versions for the pre-trained models so we can assess how well the ensemble CNN model proposed performs against the best state-of-the-art models.

6 Evaluation

6.1 Evaluation Metrics

The evaluation metrics used to assess the proposed ensemble model are accuracy, F1-Score, and the ROC AUC Score. These four metrics were selected because they are the most relevant to see how well our proposed ensemble model is behaving. Even though the confusion matrix is not an evaluation metric it helps to see how the model is classifying, so we will be using that as well.

6.1.1 Confusion Matrix

The confusion matrix is a model performance measure for classification problems. This measure is extremely useful to see how our model is classifying and it will help us understand what is happening with the model. For a binary classification problem, a confusion matrix will contain four different combinations: the true positive values, the true negative values, the false positive values, and the false negative values.

6.1.2 Accuracy

The accuracy metric will help us to see how many classifications the model got correctly out of all the classifications the model did. Below we can see how the accuracy is calculated.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives} \tag{1}$$

One thing to consider is that when the dataset is imbalanced this metric can fool how well the model it is classifying.

6.1.3 F1-Score

The F1-Score is a metric that takes into account the precision and recall metrics and combines them into one metric. The F1-Score is calculated by obtaining the harmonic mean between precision and recall, and it has a value between 0 and 1, the higher the value the better the model is at classifying between the different classes. Below we can see how the F1-Score is calculated.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2}$$

6.1.4 ROC AUC Score

The ROC AUC Score is the area under the curve of the receiving operating characteristic and this metric tells how well the model can distinguish between the Positive and Negative classes. Just like the F1-Score, the ROC AUC Score has a value between 0 and 1, the higher the value the better the model is at classifying.

6.2 Evaluation of Individual Models

In this section, we are going to evaluate how the individual models performed. In Section 5 we talk about different versions of the CNN model and with those different versions we implemented different versions of pre-trained models, so we are going to evaluate first the same versions of all models against each other and then we are going to evaluate all of them. It is important to note that the metric with the most value for this project is going to be the ROC AUC Score, this is because we have a very imbalanced dataset, and as we saw in Section 6.1 the best metric to see if the models can distinguish between the classes is the ROC AUC Score, so we are going to have a special attention on how the models perform in the ROC AUC Score.

6.2.1 Version 1

The results in Figure 3 shows that all of the models have a very high accuracy of almost 80% in all version 1 models but if we watch Table 3 we can see that all of the models are predicting positive 50% of the times that an actual image is positive, which is very bad, it is not too different to flip a coin a gain the same results. This tells us that all the models of version 1 are bad, including the pre-trained models, and that is why we should take the F1-Score metric and the ROC AUC Score metric with more value at the moment of evaluation than the accuracy metric.

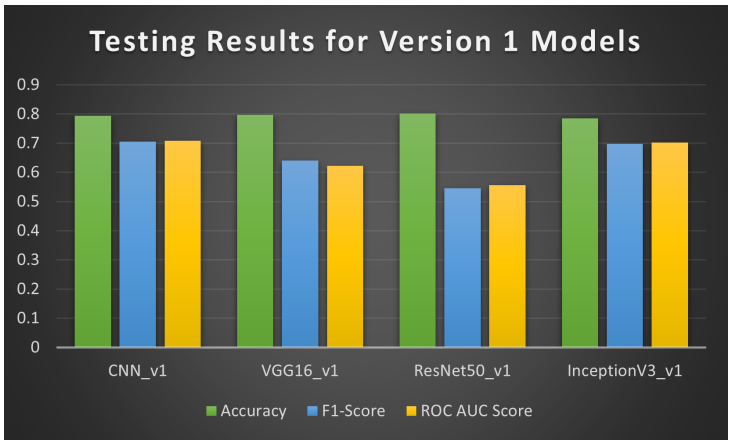

Figure 3: Testing results for version 1 models.

Table 3: Confusion Matrix Results of Version 1 Models.

Model	True Positive	True Negative	False Positive	False Negative
CNN_V1	2,133	11,586	1,710	1,858
InceptionV3_V1	2,120	11,467	1,723	1,977
VGG16_V1	1,191	12,577	2,652	867
ResNet50_V1	442	13,407	3,401	37

The best performers in this version of the models are the custom CNN that we implemented and the InceptionV3 model. The ROC AUC scores for each model were 0.708 for the CNN_v1, 0.702 for the InceptionV3_v1, 0.622 for the VGG16_v1, and 0.556 for the ResNet_v1. It is important to note that version 1 of all the models does not have any data processing to adjust for the imbalanced dataset so these results were expected, in the next versions of the model we should see an improvement.

6.2.2 Version 2

Figure 4 shows very interesting results. The InceptionV3_v2 and the VGG16_v2 models decreased the ROC AUC Score by getting 0.646 and 0.584 respectively, while the CNN_v2 and the ResNet50_v2 showed improvements by getting an ROC AUC Score of 0.734 and 0.619, respectively. These results show that the CNN architecture that is proposed can very well compete with the best pre-trained models even though it is a simple architecture.

It is important to say that these versions of the models were trained with the augmented data, one reason for the VGG16_v2 and InceptionV3_v2 doing worse than their

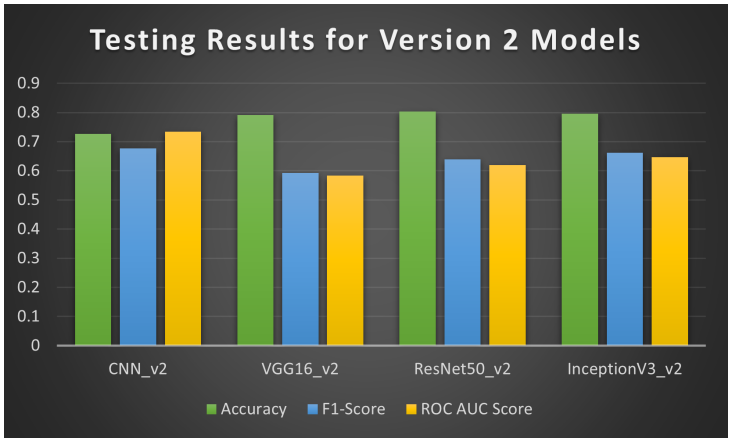

Figure 4: Testing results for version 2 models.

respective version 1 of the models is that the augmented data could not give a diverse sample to represent the positive class.

6.2.3 Version 3

The results in Figure 5 are much more expected. We can see how all the models in this version lowered their past versions' accuracy while increasing their ROC AUC Score. In Table 4 we can see the confusion matrix of all the models in this version and we can see that the positive predictions are getting a lot better in the models. One of the reasons for this improvement in all the models is because in this version we added the class weights which deals directly with the class imbalances in the dataset. The best model in this version was the InceptionV3_3 with an ROC AUC Score of 0.775, followed by the CNN_v3 model with 0.731, followed by the VGG16_v3 with 0.721, and finally the ResNet50_v3 with 0.687.

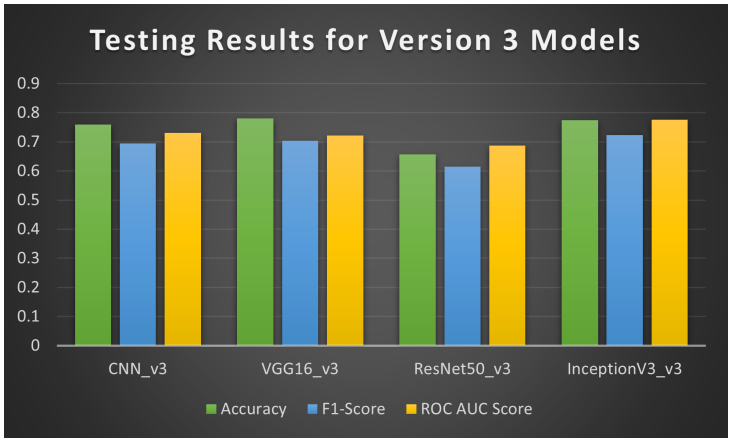

Figure 5: Testing results for version 3 models.

Table 4: Confusion Matrix Results of Version 3 Models.

Model	True Positive	True Negative	False Positive	False Negative
CNN_V1	2,618	10,505	1,225	2,939
InceptionV3_V1	2,987	10,408	856	3,036
VGG16_V1	2,367	11,118	1,476	4,947
ResNet50_V1	2,856	8,497	987	37

6.2.4 Proposed CNN Models of the Genetic Algorithm

From the results of all the CNN versions developed we could see that the best version was Version 3, so this is the version that the genetic algorithm is going to fine-tune. In Figure 6 we can see the best three models of the last generation of the genetic algorithm. The results for the CNN models that the genetic algorithm got are very good, with an ROC AUC Score of 0.763 for the first model, 0.759 for the second model, and 0.758 for the third model. These results show that our simple proposed CNN architecture is performing well.

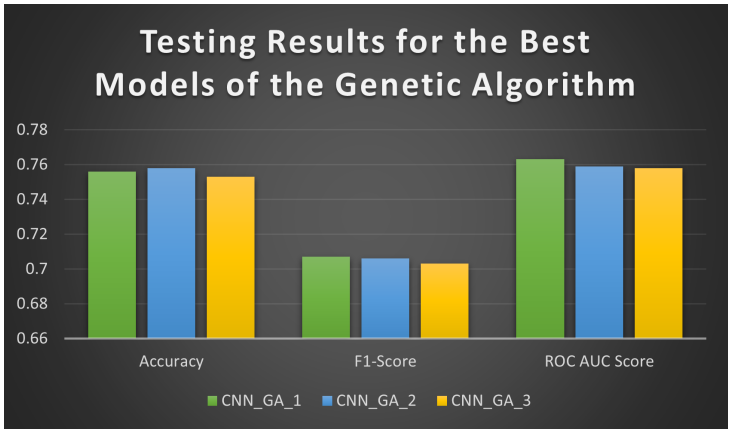

Figure 6: Testing results for CNN Genetic Algorithm models.

In Figure 7 we can see the results for all the models. The best-performing model in the ROC AUC Score metric is the InceptionV3_v3 but we can see that after the InceptionV3_v3 model, the next best-performing models are the models of the genetic algorithm.

6.3 Evaluation of Ensemble Models

The results in Table 5 show that a couple of the pre-trained ensemble models achieve higher scores than their individual models, we can also see that the proposed CNN ensemble model achieves an ROC AUC Score of 0.773, the third-best score of all the ensemble models, demonstrating that a simple CNN model fine-tune with a genetic algorithm can compete in performance to state-of-the-art pre-trained models.

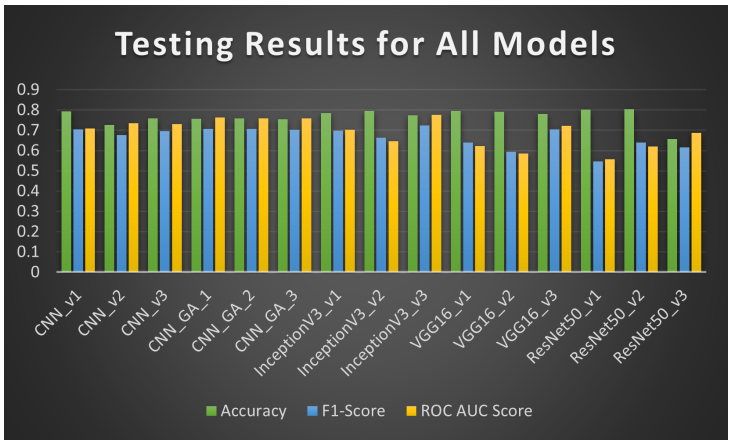

Figure 7: Testing results for all individual models.

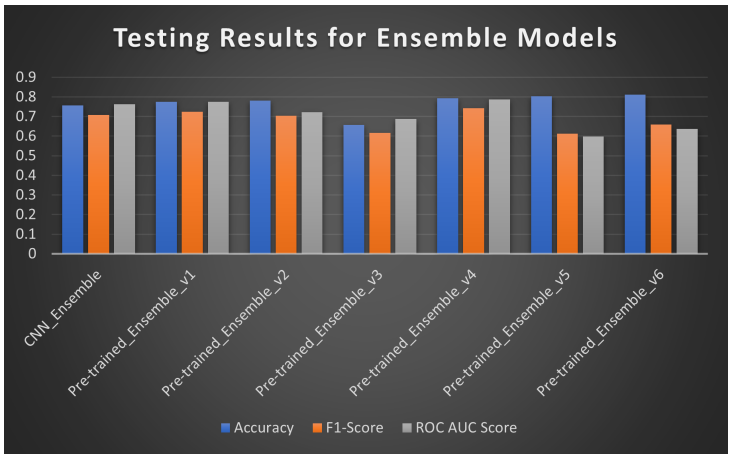

Figure 8: Testing results for ensemble models.

Table 5: Confusion Matrix Results of Ensemble Models.

Model	Accuracy	F1-Score	ROC AUC Score
CNN_Ensemble	75.6%	0.707	0.773
Pre-trained_Ensemble_V1	77.4%	0.723	0.775
Pre-trained_Ensemble_V2	78%	0.704	0.721
Pre-trained_Ensemble_V3	65.6%	0.615	0.687
Pre-trained_Ensemble_V4	79.2%	0.741	0.787
Pre-trained_Ensemble_V5	80.3%	0.611	0.597
Pre-trained_Ensemble_V6	81.1%	0.659	0.637

6.4 Discussion

Mandrekar (2010) tells us that an ROC AUC Score of 0.7-0.8 is considered good, the proposed CNN ensemble model achieved an ROC AUC Score of 0.773 making the proposed model a good model, this means that the proposed model can differentiate very well of an individual that has Alzheimer and an individual that does not have Alzheimer. The proposed model not only did a good job with the task of classifying Alzheimer's but also achieved very close scores to models that have been previously trained with datasets of millions of photos and overall, it was the third-best ensemble model.

7 Conclusion and Future Work

In this project, we assessed a custom CNN ensemble model with a simple architecture against state-of-the-art pre-trained ensemble models. Several versions of this CNN model were implemented so that the best custom CNN model could be fine-tuned by a genetic algorithm, getting the best possible parameters that the CNN model could have. A selection of the three best CNN models of the last generation of the genetic algorithm was selected to create an ensemble model to find out if it could compete against state-of-the-art pre-trained ensemble models to detect Alzheimer's in an individual. After evaluating all the models, we could see how the proposed CNN ensemble model could not only compete but also be with the best ensemble models for this task. Overall, the proposed ensemble model demonstrated, with an ROC AUC Score of 0.773, to be a good model and it was the third-best ensemble model out of all the models. Even though the proposed model is good it can be improved, one thing that can be done to improve the model is to introduce more diversity in the ensemble models, the proposed ensemble model contained the three best models by a genetic algorithm, but the three models had the same architecture, which limits the diversity of the ensemble model. For future work an implementation of three different custom CNN architectures can be done and pass every architecture to the genetic algorithm to find the best possible model and make an ensemble of the best models of the three different architectures, this will introduce some diversity in the ensemble model, and it will make the model more robust. Another thing that can be done in the future is to try to implement this proposed ensemble model to a different classification problem to see how it would perform.

References

Abed, M. T., Fatema, U., Nabil, S. A., Alam, M. A. and Reza, M. T. (2020). Alzheimer's disease prediction using convolutional neural network models leveraging pre-existing architecture and transfer learning, *2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, IEEE, pp. 1–6.

Ebrahim, D., Ali-Eldin, A. M., Moustafa, H. E. and Arafat, H. (2020). Alzheimer disease early detection using convolutional neural networks, *2020 15th international conference on computer engineering and systems (ICCES)*, IEEE, pp. 1–6.

Francis, A. and Pandian, I. A. (2021). Early detection of alzheimer's disease using ensemble of pre-trained models, *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, IEEE, pp. 692–696.

Islam, J. and Zhang, Y. (2018). Brain mri analysis for alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks, *Brain informatics* **5**: 1–14.

Jabason, E., Ahmad, M. O. and Swamy, M. (2019a). Classification of alzheimer's disease from mri data using an ensemble of hybrid deep convolutional neural networks, *2019 IEEE 62nd international Midwest symposium on circuits and systems (MWSCAS)*, IEEE, pp. 481–484.

Jabason, E., Ahmad, M. O. and Swamy, M. (2019b). Hybrid feature fusion using rnn and pre-trained cnn for classification of alzheimer's disease (poster), *2019 22th International Conference on Information Fusion (FUSION)*, IEEE, pp. 1–4.

Kang, W., Lin, L., Zhang, B., Shen, X., Wu, S., Initiative, A. D. N. et al. (2021). Multi-model and multi-slice ensemble learning architecture based on 2d convolutional neural networks for alzheimer's disease diagnosis, *Computers in Biology and Medicine* **136**: 104678.

Khagi, B., Lee, B., Pyun, J.-Y. and Kwon, G.-R. (2019). Cnn models performance analysis on mri images of oasis dataset for distinction between healthy and alzheimer's patient, *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, IEEE, pp. 1–4.

Khoei, T. T., Labuhn, M. C., Caleb, T. D., Hu, W. C. and Kaabouch, N. (2021). A stacking-based ensemble learning model with genetic algorithm for detecting early stages of alzheimer's disease, *2021 IEEE International Conference on Electro Information Technology (EIT)*, IEEE, pp. 215–222.

Loddo, A., Buttau, S. and Di Ruberto, C. (2022). Deep learning based pipelines for alzheimer's disease diagnosis: a comparative study and a novel deep-ensemble method, *Computers in biology and medicine* **141**: 105032.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment, *J. Thorac. Oncol.* **5**(9): 1315–1316.

Oktavian, M. W., Yudistira, N. and Ridok, A. (2022). Classification of alzheimer's disease using the convolutional neural network (cnn) with transfer learning and weighted loss, *arXiv preprint arXiv:2207.01584* .

Sadat, S. U., Shomee, H. H., Awwal, A., Amin, S. N., Reza, M. T. and Parvez, M. Z. (2021). Alzheimer's disease detection and classification using transfer learning technique and ensemble on convolutional neural networks, *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, pp. 1478–1481.